## REMARKS/ARGUMENTS

Claims 1, 2 and 25 are pending in the present application and were amended. Claims 9, 10, 17 and 18 were canceled. No claims were added. Reconsideration is respectfully requested in view of the above amendments and the following comments.

In this Amendment, Applicants have amended claims 1, 2 and 25 and canceled claims 9, 10, 17 and 18 from further consideration in this application. Applicants are not conceding that the subject matter encompassed by claims 1, 2, 9, 10, 17, 18 and 25 prior to this Amendment is not patentable over the art cited by the Examiner. Claims 1, 2 and 25 were amended and claims 9, 10, 17 and 18 were canceled solely to facilitate expeditious prosecution of the remaining claims. Applicants respectfully reserve the right to pursue claims, including the subject matter encompassed by claims 1, 2, 9, 10, 17, 18 and 25 as presented prior to this Amendment and additional claims in one or more continuing applications.

### I. 35 U.S.C. § 102, Anticipation

The Examiner has rejected claims 1-2, 9-10, 17-18, and 25 under 35 U.S.C. § 102(b) as being anticipated by Intel, "Intel 1A-64 Architecture Software Developer's Manual", Revision 1.1, Vol. 4, No. 245320-002, 7-2001, hereinafter "Intel". This rejection is respectfully traversed.

Claims 9, 10, 17 and 18 have been canceled. Therefore, the rejection with respect to those claims is now moot.

In rejecting the claims, the Examiner states with respect to claims 1, 2 and 25:

> Given the broadest reasonable interpretation of followed claims in light of the specification.
>
> As per Claim 1: Intel reference has 8 sections. Intel discloses,
>
> ***A method in a data processing system for monitoring execution of instructions, the method comprising: executing a program*** (See Figure 6-2 in p. 4 of sec. 6);
>
> ***Identifying a routine that is used more than a threshold number of times*** (See sec. 6.1.2 "profiling", refer to "performance monitor counts have to be associated with program locations ( *'identifying a routine'*)". Further more see sec. 6.1.1.2, 6.1.1.3 (p.6-3), sec. 6, p. 6, see event counter; The profiling' counter events are calculating based on times) ***during execution of the program as a routine of interest*** (See sec. 6.1.1.3, particularly, see its second paragraph. Intel discloses an identifying of a routine such as a benchmark is tested with different threshold values for identifying the performance "knee");
>
> ***responsive to identifying the routine of interest during execution of the program, dynamically associating instructions in the identified routine of interest with a set of performance indicators*** (sec. 6, p.7, i.e. triggers on events shown in table 6-2, see in the near end of the page, "registers indicate to...", see table 6.3, p. 10-11, and sec. 6, p. 13, "PMC/PMD register assignments for each monitoring

feature...") ***to form a modified routine*** (See sec. 6, p. 5, "are interesting identifying performance bottlenecks and relating them back to their source code": *identifying a routine of interest during execution of a program; then see "code instrumentation": in p. 26 of sec 6* ), ***wherein the set of performance indicators comprises one of a set of performance indicators comprising one or more bits located in fields within the instructions*** (See Figure 6-5, that detects indicators as instructions instrumented in the IA-64 instruction execution. These instructions are seen in sec. 7, such as instruction PIPELINE - FLUSH. Also see "performance monitor events, event counters, seen in sec. 6.1.2.2, and 6.1.2.3, p. 3, or program counter sampling for identifying hot spot, see in sec. 6, p.6 - Note: See a performance counter "monitor ++" that is implemented in a program shown in sec. 7, p.25, if take performance counters as performance indicators then each of these counter comprising 32-bits) ***and a set of performance indicators comprising metadata located in a shadow memory*** (Each of instructions shown in the sec. 7 is associated with associated with event code, and registers such as PMC/PMD (See table 6.3, p. 10-11, and sec. 6, p. 13, "PMC/PMD register assignments for each monitoring feature..."—Note if take the performance counter "monitor ++" in the program as shown in sec. 7, p. 25, then the data of this instruction is generated and stored in a shadow of a branch predict instruction (i.e. *metadata*) (e.g. see sec. 7, p. 33, the definition in INST-ACESS-CYCLE)), ***and wherein the set of performance indicators identify that the instructions are to be monitored***; (For example, monitoring cache; or see sec. 6.2.2 for setting maximum per-cycle event increment, etc); ***and***
***responsive to execution of an instruction in the modified routine*** (i.e. the routine contains hotspot results by profiling) ***during continued execution of the program, incrementing a counter*** (i.e., the performance counters. For example, see sec. 6, Figure 6-5, p. 7), ***wherein the counter provides a value identifying a number of times that the instruction in the modified routine is executed*** (e.g. sec. 6.2.2, p.16 of sec. 6).

As per Claim 2: Intel discloses, ***The method of claim 1 further comprising: associating instructions in a second routine of interest with a second set of indicators to form a second modified routine*** (Intel discloses a program that has many routines, and each of routine in monitored); ***and responsive to execution of an instruction in the second modified routine, incrementing a second counter*** (See sec. 6, Figure 6-5, p. 7).

As per Claim 25: see rationale addressed in the rejection of claims 1-2 above.

Office Action dated April 9, 2008, pages 4-6.

Amended claim 1 is as follows:

1. A method in a data processing system for monitoring execution of instructions, the method comprising:

executing a program;

identifying a routine that is used more than a threshold number of times during execution of the program as a routine of interest;

responsive to identifying the routine of interest during execution of the program, dynamically associating instructions in the identified routine of interest with a set of performance indicators to form a modified routine having the instructions, wherein the set of performance indicators comprises one or more bits located in fields within the instructions in the modified routine, and wherein the set of performance indicators identifies that the instructions in the modified routine are to be monitored; and

responsive to execution of an instruction of the instructions in the modified routine during continued execution of the program, incrementing a counter, wherein the counter provides a value identifying a number of times that the instruction of the instructions in the modified routine is executed.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

In the present case, Intel does not identically disclose or suggest every element of the claimed invention, arranged as they are in the claims, and, accordingly, does not anticipate the claims. With respect to claim 1, for example, Intel does not disclose or suggest the features of "identifying a routine that is used more than a threshold number of times during execution of the program as a routine of interest," and "responsive to identifying the routine of interest during execution of the program, dynamically associating instructions in the identified routine of interest with a set of performance indicators to form a modified routine having the instructions, wherein the set of performance indicators comprises one or more bits located in fields within the instructions in the modified routine, and wherein the set of performance indicators identifies that the instructions in the modified routine are to be monitored."

Initially, Intel does not teach or suggest "identifying a routine that is used more than a threshold number of times during execution of the program as a routine of interest." The Office Action cites to sections 6.12, 6.1.1.2 and 6.1.1.3 of Intel as allegedly teaching this feature. Applicants respectfully disagree. Section 6.12 relates to profiling to identify performance bottlenecks, and mentions program counter sampling, miss event address sampling and event qualifications. Section 6.1.1.2 describes incrementing a counter for single occurrence events. Neither of these sections relates to identifying a routine that is used more than a threshold number of times. Section 6.1.1.3 describes running a benchmark program using different threshold values in various runnings in order to create a histogram. In contradistinction, claim 1 recites identifying a routine that is used more than a threshold number of times.

Thus, although Intel may teach running a program multiple times using different threshold values, Intel does not teach or suggest identifying a program that is used more than a threshold number of times. Therefore, Intel fails to teach or in any way suggest the feature of "identifying a routine that is used more than a threshold number of times during execution of the program as a routine of interest" as recited in claim 1, and does not anticipate claim 1 for this reason.

Intel also does not teach or suggest "responsive to identifying the routine of interest during execution of the program, dynamically associating instructions in the identified routine of interest with a set of performance indicators to form a modified routine having the instructions, wherein the set of performance indicators comprises one or more bits located in fields within the instructions in the modified routine, and wherein the set of performance indicators identifies that the instructions in the modified routine are to be monitored." as recited in amended claim 1. Intel, in fact, does not disclose or suggest a set of performance indicators that comprises one or more bits located in fields within the instructions as recited in amended claim 1. The Office Action appears to equate a set of indicators with a set of elements that cause problems such as bottlenecks and so forth. However, amended claim 1 now clearly specifies that performance indicators comprise one or more bits located in fields within the instructions.

Intel merely teaches that events occur and that those events are counted. In contradistinction, claim 1 recites that responsive to identifying a routine of interest, instructions are dynamically associated with the performance indicators, forming a modified routine, and further, that the performance indicators are comprised of one or more bits located in fields within the instructions. Nowhere does Intel teach or even hint at identifying a routine and a set of instructions and then dynamically modifying the instructions once they have been identified to include performance indicators that comprise one or more bits located in fields within the instructions. Again, Intel merely teaches counting events when events occur.

Thus, Intel also fails to teach or suggest the feature of "responsive to identifying the routine of interest during execution of the program, dynamically associating instructions in the identified routine of interest with a set of performance indicators to form a modified routine having the instructions, wherein the set of performance indicators comprises one or more bits located in fields within the instructions in the modified routine, and wherein the set of performance indicators identifies that the instructions in the modified routine are to be monitored", and does not anticipate claim 1 for this reason as well.

Therefore, for at least all the above reasons, Applicants submit that Intel fails to anticipate claim 1, as Intel fails to teach each and every feature of claim 1. Claim 1, accordingly, patentably distinguishes over Intel in its present form.

Claim 2 depends from and further restricts claim 1, and patentably distinguishes over Intel at least by virtue of its dependency. Claim 25 recites similar subject matter as claims 1 and 2 and is also not anticipated by Intel for similar reasons as discussed above with respect to claim 1.

Therefore, the rejection of claims 1-2, 9-10, 17-18, and 25 under 35 U.S.C. § 102 has been overcome.

## II. Conclusion

It is respectfully urged that the subject application is patentable over Intel and is now in condition for allowance. It is, accordingly, respectfully requested that the Examiner so find and issue a Notice of Allowance in due course.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: July 9, 2008

Respectfully submitted,

/Gerald H. Glanzman/

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants